CLAIMS

What is claimed is:

1. A method for buffering data in a computer graphics pipeline, comprising:
(a) producing graphics floating point data in a graphics pipeline;
(b) operating on the graphics floating point data in the graphics pipeline; and
(c) storing the graphics floating point data to a buffer in the graphics pipeline;
(d) wherein the graphics floating point data is read and stored in an unclamped format for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

2. The method as recited in claim 1, wherein the graphics floating point data includes fragment data.

3. The method as recited in claim 2, wherein the fragment data is received from a rasterizer.

4. The method as recited in claim 2, wherein the fragment data includes color data.

5. The method as recited in claim 2, wherein the fragment data includes depth data.

6. The method as recited in claim 1, wherein the graphics floating point data is only constrained by an underlying data type.

7. The method as recited in claim 1, wherein the buffer serves as a texture map.

8. A computer program product for buffering data in a computer graphics pipeline, comprising:

(a) computer code for producing graphics floating point data in a graphics pipeline;

(b) computer code for operating on the graphics floating point data in the graphics pipeline; and

(c) computer code for storing the graphics floating point data to a buffer in the graphics pipeline;

(d) wherein the graphics floating point data is read and stored in an unclamped format for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

9. The computer program product as recited in claim 8, wherein the graphics floating point data includes fragment data.

10. The computer program product as recited in claim 9, wherein the fragment data is received from a rasterizer.

11. The computer program product as recited in claim 9, wherein the fragment data includes color data.

12. The computer program product as recited in claim 9, wherein the fragment data includes depth data.

13. The computer program product as recited in claim 8, wherein the graphics floating point data is only constrained by an underlying data type.

14. The computer program product as recited in claim 8, wherein the buffer serves as a texture map.

15. A system for buffering data in a computer graphics pipeline, comprising:
(a) logic for producing graphics floating point data in a graphics pipeline;
(b) logic for operating on the graphics floating point data in the graphics pipeline; and
(c) logic for storing the graphics floating point data to a buffer in the graphics pipeline;
(d) wherein the graphics floating point data is read and stored in an unclamped format for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

16. A buffering apparatus in a computer graphics pipeline, comprising:
(a) a buffer capable of storing graphics floating point data in a graphics pipeline;
(b) wherein the graphics floating point data is stored in an unclamped format for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

17. A system for buffering data in a computer graphics pipeline, comprising:
(a) means for producing graphics floating point data in a graphics pipeline;
(b) means for operating on the graphics floating point data in the graphics pipeline; and
(c) means for storing the graphics floating point data to a buffer in the graphics pipeline;
(d) wherein the graphics floating point data is read and stored in an unclamped format for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

18. A method for buffering data in a computer graphics pipeline, comprising:
(a) producing graphics floating point data in a graphics pipeline;
(b) operating on the graphics floating point data in the graphics pipeline; and

(c) storing the graphics floating point data to a buffer in the graphics pipeline;

(d) wherein the buffer serves as a texture map.

19. A buffering apparatus in a computer graphics pipeline, comprising:

(a) a buffer capable of storing graphics floating point data in a graphics pipeline;

(b) wherein the buffer serves as a texture map.

20. A method for buffering data during multi-pass rendering in a computer graphics pipeline, comprising:

(a) operating on graphics floating point data during a rendering pass in a graphics pipeline;

(b) reading the graphics floating point data from a buffer during the rendering pass in the graphics pipeline;

(c) storing the graphics floating point data to the buffer during the rendering pass in the graphics pipeline; and

(d) repeating (a) – (c) during additional rendering passes.

21. The method as recited in claim 20, wherein the operating includes deferred shading.

22. A method for buffering data in a computer graphics pipeline, comprising:

(a) producing graphics floating point data in a graphics pipeline;

(b) packing the graphics floating point data in the graphics pipeline; and

(c) storing the graphics floating point data to a buffer in the graphics pipeline.

23. A method for buffering data in a computer graphics pipeline, comprising:

(a) producing graphics floating point data in a graphics pipeline;

(b) unpacking the graphics floating point data in the graphics pipeline; and

(c) operating on the unpacked graphics floating point data in the graphics pipeline.

24. A method for buffering data in a computer graphics pipeline, comprising:

(a) operating on graphics floating point data in a graphics pipeline;

(b) producing the graphics floating point data in the graphics pipeline; and

(c) storing the graphics floating point data to a buffer in the graphics pipeline;

(d) wherein the graphics floating point data is read and stored in an unclamped format for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.